L Number	Hits	Search Text	DB	Time stamp
1	1254	(CAD or (computer adj (aided or assisted))) and (file$1 or object$1) and (update$1 or modif$2 or translate$1) and distribut$3 and hierarch$4 and format$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/18 20:35
2	1238	((CAD or (computer adj (aided or assisted))) and (file$1 or object$1) and (update$1 or modif$2 or translate$1) and distribut$3 and hierarch$4 and format$1) and ((different or incompatible or ("not" adj supported) with format$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/18 20:38
3	295	((CAD or (computer adj (aided or assisted))) and (file$1 or object$1) and (update$1 or modif$2 or translate$1) and distribut$3 and hierarch$4 and format$1) and (((different or incompatible or ("not" adj supported)) with format$1))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/18 20:39
4	37	(((CAD or (computer adj (aided or assisted))) and (file$1 or object$1) and (update$1 or modif$2 or translate$1) and distribut$3 and hierarch$4 and format$1) and (((different or incompatible or ("not" adj supported)) with format$1))) and (over$1write$1 or super$1cede$1)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/18 20:45
-	2	4864497.pn.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/18 20:31
-	57	4864497.uref.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2003/08/18 19:10

	Document ID	Issue Date	Title	Current OR
1	US 6550057 B1	20030415	Piecemeal retrieval in an information services patterns environment	717/126
2	US 6393426 B1	20020521	Method for modeling, storing and transferring data in neutral form	707/100
3	US 6216262 B1	20010410	Distributed processing	717/149
4	US 6212557 B1	20010403	Method and apparatus for synchronizing upgrades in distributed network data processing systems	709/221
5	US 6208992 B1	20010327	Information system and process for storing data therein	707/100
6	US 6138202 A	20001024	Object space manager circuit for obtaining addresses of object headers	711/101
7	US 6134304 A	20001017	General analysis system	379/100.05
8	US 6085233 A	20000704	System and method for cellular network computing and communications	709/216
9	US 6047296 A	20000404	Comprehensive method of resolving nested forward references in electronic data streams within defined resolution scopes	715/513

uref reference on 4864497 patent

	Document ID	Issue Date	Title	Current OR
10	US 5978577 A	19991102	Method and apparatus for transaction processing in a distributed database system	707/10
11	US 5937402 A	19990810	System for enabling access to a relational database from an object oriented program	707/4
12	US 5870733 A	19990209	Automated system and method for providing access data concerning an item of business property	707/2
13	US 5856828 A	19990105	System of relational entities for object-oriented computer-aided geometric design	345/420
14	US 5850535 A	19981215	Roll-back during regeneration on a computer-aided design system	703/2
15	US 5845296 A	19981201	Method and apparatus for implementing segmented arrays in a database	707/205
16	US 5842213 A	19981124	Method for modeling, storing, and transferring data in neutral form	707/100
17	US 5832517 A	19981103	System logging using embedded database	707/202
18	US 5832266 A	19981103	Encapsulation of applications for inclusion within an object management facility environment	709/316

	Document ID	Issue Date	Title	Current OR
19	US 5826270 A	19981020	Methods and systems for client or customer-site transaction processing in a distributed database system	707/10
20	US 5822781 A	19981013	Sector-based storage device emulator having variable-sized sector	711/171
21	US 5819282 A	19981006	Database generator	707/103R
22	US 5805869 A	19980908	System for computer with interface and scripting systems cooperating in interrated fashion by sharing frame objects of common unified data structure stored in object system	345/713
23	US 5794042 A	19980811	File management apparatus permitting access to portions of a file by specifying a data structure identifier and data elements	717/101
24	US 5740423 A	19980414	System and method for accessing distributed data on a plurality of databases	707/10
25	US 5732271 A	19980324	Data processing system and method for processing an object oriented development environment employing property inheritance using prototypical objects	709/316
26	US 5717922 A	19980210	Method and system for management of logical links between document elements during document interchange	707/100
27	US 5680586 A	19971021	Method and system for storing and accessing user-defined attributes within a data processing system	703/27
28	US 5659751 A	19970819	Apparatus and method for dynamic linking of computer software components	709/332

	Document ID	Issue Date	Title	Current OR
29	US 5627949 A	19970506	System of relational entities for object-oriented computer-aided geometric design	345/420
30	US 5623419 A	19970422	Modeling of multi-disciplinary signals	716/5
31	US 5615333 A	19970325	Integration testing method for object-oriented software	714/38
32	US 5588141 A	19961224	System for executing different functions associated with different contexts corresponding to different screen events based upon information stored in unified data structure	709/315
33	US 5586317 A	19961217	Method and apparatus for implementing I/O in a frame-based computer system	709/320
34	US 5581672 A	19961203	System of relational entities for object-oriented computer-aided geometric design	345/420
35	US 5561802 A	19961001	Method for managing programs with attribute information and developing loaded programs	717/136
36	US 5553127 A	19960903	System and method for processing and analyzing telephone calls in a telephone exchange	379/201.02
37	US 5542085 A	19960730	Method for producing program modules	707/6
38	US 5542078 A	19960730	Object oriented data store integration environment for integration of object oriented databases and non-object oriented data facilities	707/101
39	US 5495603 A	19960227	Declarative automatic class selection filter for dynamic file reclassification	707/200

	Document ID	Issue Date	Title	Current OR
40	US 5488732 A	19960130	Automatic support tool generating system based on a knowledge acquisition tool and a database	707/104.1
41	US 5434118 A	19950718	Catalytically active gel and a process for its preparation	502/242
42	US 5355476 A	19941011	File update apparatus for generating a matrix representing a subset of files and the update correspondence between directories and files	707/1
43	US 5315709 A	19940524	Method and apparatus for transforming objects in data models	707/6
44	US 5313636 A	19940517	Mosaic objects and method for optimizing object representation performance in an object-oriented representation system	707/1
45	US 5295081 A	19940315	Concurrent interactive wire editing	716/11
46	US 5280575 A	19940118	Apparatus for cell format control in a spread sheet	715/504
47	US 5265206 A	19931123	System and method for implementing a messenger and object manager in an object oriented programming environment	709/316
48	US 5210868 A	19930511	Database system and matching method between databases	707/5
49	US 5175810 A	19921229	Tabular data format	715/509
50	US 5133075 A	19920721	Method of monitoring changes in attribute values of object in an object-oriented database	707/201

	Document ID	Issue Date	Title	Current OR
51	US 5129083 A	19920707	Conditional object creating system having different object pointers for accessing a set of data structure objects	707/103R
52	US 5129081 A	19920707	System for processing data using logic language	707/1
53	US 5123103 A	19920616	Method and system of retrieving program specification and linking the specification by concept to retrieval request for reusing program parts	707/5
54	US 5115504 A	19920519	Information management system	707/100
55	US 5101493 A	19920331	Digital computer using data structure including external reference arrangement	707/1
56	US 5101345 A	19920331	Method of filing stapled documents with a staple relationship involving one or more application programs	707/1
57	US 5093779 A	19920303	Computer file system	707/1

PORTAL
THE ACM DIGITAL LIBRARY

> home > about > feedback > login
US Patent & Trademark Office

Beta Try the *new* Portal design
Give us your opinion after using it.

Search Results

Search Results for: **[CAD and (file or object) and (update or modify or translate) and distributed and overwrite and (hierarchy or hierarchical) and format]**
Found **22** of **120,398 searched.**

Search within Results

GO > Advanced Search > Search Help/Tips

Sort by: Title Publication Publication Date Score Binder

Results 1 - 20 of 22 short listing

Prev Page 1 2 Next Page

1 From VHDL to efficient and first-time-right designs: a formal approach 80%
Peter F. A. Middelhoek , Sreeranga P. Rajan
ACM Transactions on Design Automation of Electronic Systems (TODAES) April 1996
Volume 1 Issue 2
In this article we provide a practical transformational approach to the synthesis of correct synchronous digital hardware designs from high-level specifications. We do this while taking into account the complete life cycle of a design from early prototype to full custom implementation. Besides time-to-market, both flexibility with respect to target architecture and efficiency issues are addressed by the methodology. The utilization of user-selected behavior-preserving transformation steps e ...

2 Cooperative transaction hierarchies: transaction support for design applications 80%
Marian H. Nodine , Stanley B. Zdonik
The VLDB Journal — The International Journal on Very Large Data Bases July 1992
Volume 1 Issue 1
Traditional atomic and nested transactions are not always well-suited to cooperative applications, such as design applications. Cooperative applications place requirements on the database that may conflict with the serializability requirement. They require transactions to be long, possibly nested, and able to interact with each other in a structured way. We define a transaction framework, called a *cooperative transaction hierarchy,* that allows us to relax the requirement for atomic, seria ...

3 System facilities for CAD databases 77%
Charles M. Eastman
Proceedings of the seventeenth design automation conference on Design automation June 1980
In this paper, an attempt is made to lay out the special needs of design databases, as compared to

the facilities provided in conventional database systems now commercially available. The paper starts from a point of commonality and focusses on the limitations and shortcomings commonly found in current database systems. It is impossible and unwise to make universal statements about DBMS capabilities. Instead, the goal is to identify those special features that, by their capability, provide ...

4 Continuous program optimization: A case study 77%

Thomas Kistler , Michael Franz

ACM Transactions on Programming Languages and Systems (TOPLAS) July 2003
Volume 25 Issue 4

Much of the software in everyday operation is not making optimal use of the hardware on which it actually runs. Among the reasons for this discrepancy are hardware/software mismatches, modularization overheads introduced by software engineering considerations, and the inability of systems to adapt to users' behaviors.A solution to these problems is to delay code generation until load time. This is the earliest point at which a piece of software can be fine-tuned to the actual capabilities of the ...

5 PCLOS: a critical review 77%

A. Paepcke

ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages and applications September 1989
Volume 24 Issue 10

This paper uses the persistent object system PCLOS to survey some problems and benefits of object persistence. The system is analyzed along several relevant dimensions. PCLOS provides object persistence for an object-oriented language. The insights gained on desirable and detrimental components of the system are presented. The intent is to outline some of the expected and unexpected problems encountered in the construction of support for object persistence.

6 The design and implementation of K: a high-level knowledge-base programming language of OSAM*.KBMS 77%

Yuh-Ming Shyy , Javier Arroyo , Stanley Y.W. Su , Herman Lam

The VLDB Journal — The International Journal on Very Large Data Bases August 1996
Volume 5 Issue 3

The OSAM*.KBMS is a knowledge-base management system, or the so-called next-generation database management system, for non-traditional data/knowledge-intensive applications. In order to define, query, and manipulate a knowledge base, as well as to write codes to implement any application system, we have developed an object-oriented knowledge-base programming language called K to serve as the high-level interface of OSAM*.KBMS. This paper presents the design of K, its implementation, and its supp ...

7 Computational hypertext in biological modelling 77%

J. L. Schnase , J. J. Leggett

Proceedings of the second annual ACM conference on Hypertext November 1989

This paper describes an application of hypertext to a biological research problem. An individual energetics model for Cassin's Sparrow was developed in which the computations and intellectual activities associated with each phase of the research were performed within an integrated hypertext environment. The study demonstrates the effectiveness of computational hypertext in

meeting the personal information management requirements of individual researchers in the natural sciences and its abil ...

8 State trees as structured finite state machines for user interfaces 77%
James Rumbaugh
Proceedings of the 1st annual ACM SIGGRAPH symposium on User Interface Software January 1988
State trees are a technique for specifying the control of an interactive system by organizing states into trees to define shared structure and behavior. The tree structure permits inheritance of state information, event traps, and entry and exit actions from states to substates, thereby sharing information and reducing the amount of code to be written. An interface can be restructured by moving entire subtrees as modules. State trees separate the recognition of commands from their implement ...

9 Reconfigurable computing: a survey of systems and software 77%
Katherine Compton , Scott Hauck
ACM Computing Surveys (CSUR) June 2002
Volume 34 Issue 2
Due to its potential to greatly accelerate a wide variety of applications, reconfigurable computing has become a subject of a great deal of research. Its key feature is the ability to perform computations in hardware to increase performance, while retaining much of the flexibility of a software solution. In this survey, we explore the hardware aspects of reconfigurable computing machines, from single chip architectures to multi-chip systems, including internal structures and external coupling. W ...

10 Synthesis, Verification and Test: Timing verification of dynamically reconfigurable logic for the xilinx virtex FPGA series 77%
Ian Robertson , James Irvine , Patrick Lysaght , David Robinson
Proceedings of the 2002 ACM/SIGDA tenth international symposium on Field-programmable gate arrays February 2002
This paper reports on a method for extending existing VHDL design and verification software available for the Xilinx Virtex series of FPGAs. It allows the designer to apply standard hardware design and verification tools to the design of dynamically reconfigurable logic (DRL). The technique involves the conversion of a dynamic design into multiple static designs, suitable for input to standard synthesis and APR tools. For timing and functional verification after APR, the sections of the design c ...

11 Special Session on Design Paradigms: Interoperability as a design issue in C++ based modeling environments 77%
Frederic Doucet , Rajesh Gupta , Masato Otsuka , Patrick Schaumont , Sandeep Shukla
Proceedings of the 14th international symposium on Systems synthesis September 2001
The increasing heterogeneity and complexity of VLSI systems has made the use of C++ popular for building simulation and synthesis models at higher levels of abstraction. Currently, there are several different embodiments of C++ based environments, mostly in the form of hardware modeling libraries built on top of C++. However, the *semantic gap*between hardware modeling concepts, and the software programming language constructs, poses several isues which require critical examination. In this ...

12 The making of VIVID: a software engineering perspective 77%
J. B. Rosenberg
24th ACM/IEEE conference proceedings on Design automation conference October 1987
This paper is about the software engineering facets of the making of a large programming systems product for symbolic VLSI CAD called VIVID. Issues such as how teams were organized, how conceptual integrity was maintained, and portability are discussed. Specific tools are also described that were built to aid in the development of VIVID. Methods are explored that aid in managing large systems with many directories and huge numbers of files. Finally, techniques for debugging such a large sys ...

13 Chimera: hypermedia for heterogeneous software development enviroments 77%
Kenneth M. Anderson , Richard N. Taylor , E. James Whitehead
ACM Transactions on Information Systems (TOIS) July 2000
Volume 18 Issue 3
Emerging software development environments are characterized by heterogeneity: they are composed of diverse object stores, user interfaces, and tools. This paper presents an approach for providing hypermedia services in this heterogeneous setting. Central notions of the approach include the following: anchors are established with respect to interactive views of objects, rather than the objects themselves; composable, n-ary links can be established between a ...

14 The file system of an integrated local network 77%
Paul J. Leach , Paul H. Levine , James A. Hamilton , Bernard L. Stumpf
Proceedings of the 1985 ACM thirteenth annual conference on Computer Science March 1985
The distributed file system component of the DOMAIN system is described. The DOMAIN system is an architecture for networks of personal workstations and servers which creates an integrated distributed computing environment. The distinctive features of the file system include: objects addressed by unique identifiers (UIDs); transparent access to objects, regardless of their location in the network; the abstraction of a single level store for accessing all objects; and the layering of a networ ...

15 Automated extraction of SPICE circuit models from symbolic gate matrix layout with pruning 77%
R. D. Freeman , S. M. Kang , C. G. Lin-Hendel , M. L. Newby
Proceedings of the 23rd ACM/IEEE conference on Design automation July 1986
VLSI designers have made extensive use of SPICE simulation to analyze timing-critical circuits such as critical paths and clock distribution networks. Rigorous modeling of resistive and capacitive parasitics and transistors is required for these timing-critical circuits. Unfortunately the conventional circuit extractors have been unable to model wiring resistance and extracting the essential subcircuits, and therefore have required extensive manual editing. Manual editing is so complicated ...

16 NSF workshop on industrial/academic cooperation in database systems 77%
Mike Carey , Len Seligman
ACM SIGMOD Record March 1999
Volume 28 Issue 1

17 Referential integrity of links in open hypermedia systems 77%
Hugh C. Davis

Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space---structure in hypermedia systems: links, objects, time and space---structure in hypermedia systems May 1998

18 Power exploration for data dominated video applications 77%
Sven Wuytack , Francky Catthoor , Lode Nachtergaele , Hugo De Man
Proceedings of the 1996 international symposium on Low power electronics and design August 1996

19 Implementing crash recovery in QuickStore: a performance study 77%
Seth J. White , David J. DeWitt
ACM SIGMOD Record , Proceedings of the 1995 ACM SIGMOD international conference on Management of data May 1995
Volume 24 Issue 2
Implementing crash recovery in an Object-Oriented Database System (OODBMS) raises several challenging issues for performance that are not present in traditional DBMSs. These performance concerns result both from significant architectural differences between OODBMSs and traditional database systems and differences in OODBMS's target applications. This paper compares the performance of several alternative approaches to implementing crash recovery in an OODBMS based on a client-server architecture. ...

20 Functionality and architecture of a cooperative database system: a vision 77%
Thomas Kirsche , Richard Lenz , Hans Schuster
Proceedings of the third international conference on Information and knowledge management November 1994
A database system fostering the cooperative usage and modification of a common data pool should provide standard database functionality (e.g. application-independent correctness criteria and data modelling) plus means for a step-wise, cooperative refinement of data over a long period of time. Key ingredients are a hierarchical organization of work, a sound data model covering cooperative uncertainly, and support for long-living cooperative processes. Furthermore, mechanisms for data passing ...

Results 1 - 20 of 22 short listing

Prev Page 1 2 Next Page

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2003 ACM, Inc.

PORTAL DIGITAL LIBR

> home > about > feedback > login
US Patent & Trademark Office

Try the *new* Portal design
Give us your opinion after using it.

Search Results

Search Results for: **[CAD and (file or object) and update and distributed and overwrite and (hierarchy or hierarchical)]**
Found **41** of **120,398 searched.**

Search within Results

GO > Advanced Search > Search Help/Tips

Sort by: Title Publication Publication Date Score Binder

Results 1 - 20 of 41 short listing

Prev Page 1 2 3 Next Page

1 Special issue on persistent object systems: Orthogonally persistent object systems 80%
Malcolm Atkinson , Ronald Morrison
The VLDB Journal — The International Journal on Very Large Data Bases July 1995
Volume 4 Issue 3
Persistent Application Systems (PASs) are of increasing social and economic importance. They have the potential to be long-lived, concurrently accessed, and consist of large bodies of data and programs. Typical examples of PASs are CAD/CAM systems, office automation, CASE tools, software engineering environments, and patient-care support systems in hospitals. Orthogonally persistent object systems are intended to provide improved support for the design, construction, maintenance, and operation o ...

2 Cooperative transaction hierarchies: transaction support for design applications 80%
Marian H. Nodine , Stanley B. Zdonik
The VLDB Journal — The International Journal on Very Large Data Bases July 1992
Volume 1 Issue 1
Traditional atomic and nested transactions are not always well-suited to cooperative applications, such as design applications. Cooperative applications place requirements on the database that may conflict with the serializability requirement. They require transactions to be long, possibly nested, and able to interact with each other in a structured way. We define a transaction framework, called a *cooperative transaction hierarchy,* that allows us to relax the requirement for atomic, seria ...

3 Adaptive, fine-grained sharing in a client-server OODBMS: a callback-based approach 80%
Markos Zaharioudakis , Michael J. Carey , Michael J. Franklin
ACM Transactions on Database Systems (TODS) December 1997
Volume 22 Issue 4

For reasons of simplicity and communication efficiency, a number of existing object-oriented database management systems are based on page server architectures; data pages are their minimum unit of transfer and client caching. Despite their efficiency, page servers are often criticized as being too retrictive when it comes to concurrency, as existing systems use pages as the minimum locking unit as well. In this paper we show how to support object-level locking in a page-server context. Sev ...

4 A survey of current object-oriented databases 80%
Mansour Zand , Val Collins , Dale Caviness
ACM SIGMIS Database February 1995
Volume 26 Issue 1
Object-oriented concepts form a good basis for the data models required for next-generation database applications such as CAD/CAE/CASE/CAM systems, knowledge-based systems, multimedia, etc. Many object-oriented databases are available commercially or are being developed by industry or academic research facilities. This paper attempts to compare some of these products using fourteen criteria. The selected criteria are major factors required for the successful design of an object-oriented database ...

5 System facilities for CAD databases 77%
Charles M. Eastman
Proceedings of the seventeenth design automation conference on Design automation June 1980
In this paper, an attempt is made to lay out the special needs of design databases, as compared to the facilities provided in conventional database systems now commercially available. The paper starts from a point of commonality and focusses on the limitations and shortcomings commonly found in current database systems. It is impossible and unwise to make universal statements about DBMS capabilities. Instead, the goal is to identify those special features that, by their capability, provide ...

6 Continuous program optimization: A case study 77%
Thomas Kistler , Michael Franz
ACM Transactions on Programming Languages and Systems (TOPLAS) July 2003
Volume 25 Issue 4
Much of the software in everyday operation is not making optimal use of the hardware on which it actually runs. Among the reasons for this discrepancy are hardware/software mismatches, modularization overheads introduced by software engineering considerations, and the inability of systems to adapt to users' behaviors.A solution to these problems is to delay code generation until load time. This is the earliest point at which a piece of software can be fine-tuned to the actual capabilities of the ...

7 PCLOS: a critical review 77%
A. Paepcke
ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages and applications September 1989
Volume 24 Issue 10
This paper uses the persistent object system PCLOS to survey some problems and benefits of object persistence. The system is analyzed along several relevant dimensions. PCLOS provides object persistence for an object-oriented language. The insights gained on desirable and

detrimental components of the system are presented. The intent is to outline some of the expected and unexpected problems encountered in the construction of support for object persistence.

8 The design and implementation of K: a high-level knowledge-base programming language of OSAM*.KBMS 77%

Yuh-Ming Shyy , Javier Arroyo , Stanley Y.W. Su , Herman Lam

The VLDB Journal — The International Journal on Very Large Data Bases August 1996

Volume 5 Issue 3

The OSAM*.KBMS is a knowledge-base management system, or the so-called next-generation database management system, for non-traditional data/knowledge-intensive applications. In order to define, query, and manipulate a knowledge base, as well as to write codes to implement any application system, we have developed an object-oriented knowledge-base programming language called K to serve as the high-level interface of OSAM*.KBMS. This paper presents the design of K, its implementation, and its supp ...

9 Synchronization and recovery in a client-server storage system 77%

E. Panagos , A. Biliris

The VLDB Journal — The International Journal on Very Large Data Bases August 1997

Volume 6 Issue 3

Client-server object-oriented database management systems differ significantly from traditional centralized systems in terms of their architecture and the applications they target. In this paper, we present the client-server architecture of the EOS storage manager and we describe the concurrency control and recovery mechanisms it employs. EOS offers a semi-optimistic locking scheme based on the multi-granularity two-version two-phase locking protocol. Under this scheme, multiple concurrent reade ...

10 Loop optimization in register-transfer scheduling for DSP-systems 77%

G. Goossens , J. Vandewlle , H. De Man

Proceedings of the 1989 26th ACM/IEEE conference on Design automation conference June 1989

In this paper, we discuss a control-flow transformation called loop folding, during the scheduling of register-transfer code for DSP-systems. Loop folding is functionally equivalent to data-path pipelining. An iterative loop-folding procedure, implemented in the CATHEDRAL II compiler, is presented. This technique may significantly improve the utilization of parallel hardware, available in a data path.

11 State trees as structured finite state machines for user interfaces 77%

James Rumbaugh

Proceedings of the 1st annual ACM SIGGRAPH symposium on User Interface Software January 1988

State trees are a technique for specifying the control of an interactive system by organizing states into trees to define shared structure and behavior. The tree structure permits inheritance of state information, event traps, and entry and exit actions from states to substates, thereby sharing information and reducing the amount of code to be written. An interface can be restructured by moving entire subtrees as modules. State trees separate the recognition of commands from their implement ...

12 Ray tracing vs. scan conversion: SaarCOR: a hardware architecture for ray tracing 77%
Jörg Schmittler , Ingo Wald , Philipp Slusallek
Proceedings of the conference on Graphics hardware 2002 September 2002
The ray tracing algorithmis well-known for its ability to generate high-quality images and its flexibility to support advanced rendering and lighting effects. Interactive ray tracing has been shown to work well on clusters of PCs and supercomputers but direct hardware support for ray tracing has been difficult to implement.In this paper, we present a new, scalable, modular, and highly efficient hardware architecture for real-time ray tracing. It achieves high performance with extremely low memor ...

13 The VMP multiprocessor: initial experience, refinements, and performance evaluation 77%
D. R. Cheriton , A. Gupta , P. D. Boyle , H. A. Goosen
ACM SIGARCH Computer Architecture News , Proceedings of the 15th Annual International Symposium on Computer architecture May 1988
Volume 16 Issue 2
VMP is an experimental multiprocessor being developed at Stanford University, suitable for high-performance workstations and server machines. Its primary novelty lies in the use of software management of the per-processor caches and the design decisions in the cache and bus that make this approach feasible. The design and some uniprocessor trace-driven simulations indicating its performance have been reported previously. In this paper, we present our initial experience with the V ...

14 Reconfigurable computing: a survey of systems and software 77%
Katherine Compton , Scott Hauck
ACM Computing Surveys (CSUR) June 2002
Volume 34 Issue 2
Due to its potential to greatly accelerate a wide variety of applications, reconfigurable computing has become a subject of a great deal of research. Its key feature is the ability to perform computations in hardware to increase performance, while retaining much of the flexibility of a software solution. In this survey, we explore the hardware aspects of reconfigurable computing machines, from single chip architectures to multi-chip systems, including internal structures and external coupling. W ...

15 BASE: using abstraction to improve fault tolerance 77%
Rodrigo Rodrigues , Miguel Castro , Barbara Liskov
ACM SIGOPS Operating Systems Review , Proceedings of the eighteenth ACM symposium on Operating systems principles October 2001
Volume 35 Issue 5
Software errors are a major cause of outages and they are increasingly exploited in malicious attacks. Byzantine fault tolerance allows replicated systems to mask some software errors but it is expensive to deploy. This paper describes a replication technique, BASE, which uses abstraction to reduce the cost of Byzantine fault tolerance and to improve its ability to mask software errors. BASE reduces cost because it enables reuse of off-the-shelf service implementations. It improves availability ...

16 Special Session on Design Paradigms: Interoperability as a design issue in C++ based modeling environments 77%

Frederic Doucet , Rajesh Gupta , Masato Otsuka , Patrick Schaumont , Sandeep Shukla
Proceedings of the 14th international symposium on Systems synthesis September 2001
The increasing heterogeneity and complexity of VLSI systems has made the use of C++ popular for building simulation and synthesis models at higher levels of abstraction. Currently, there are several different embodiments of C++ based environments, mostly in the form of hardware modeling libraries built on top of C++. However, the *semantic gap*between hardware modeling concepts, and the software programming language constructs, poses several isues which require critical examination. In this ...

17 EASE: a design support environment for the HDDL ELLA 77%
J. D. Morison , N. E. Peeling , T. L. Thorp , E. jV. Whiting
24th ACM/IEEE conference proceedings on Design automation conference October 1987
This paper describes the ELLA applications support environment - EASE. The support environment allows separate modular compilation and multi-level simulation with almost no semantic or syntatic constructs in the ELLA language. The paper describes the form that the EASE takes and the advantages this gives over other programming support environments. The weaknesses of the current EASE and future work are also discussed.

18 The making of VIVID: a software engineering perspective 77%
J. B. Rosenberg
24th ACM/IEEE conference proceedings on Design automation conference October 1987
This paper is about the software engineering facets of the making of a large programming systems product for symbolic VLSI CAD called VIVID. Issues such as how teams were organized, how conceptual integrity was maintained, and portability are discussed. Specific tools are also described that were built to aid in the development of VIVID. Methods are explored that aid in managing large systems with many directories and huge numbers of files. Finally, techniques for debugging such a large sys ...

19 Dynamic space management for user interfaces 77%
Blaine A. Bell , Steven K. Feiner
Proceedings of the 13th annual ACM symposium on User interface software and technology November 2000

20 Supporting cognitive models as users 77%
Frank E. Ritter , Gordon D. Baxter , Gary Jones , Richard M. Young
ACM Transactions on Computer-Human Interaction (TOCHI) June 2000
Volume 7 Issue 2
Cognitive models are computer programs that simulate human performance of cognitive skills. They have been useful to HCI by predicting task times, by assisting users, and by acting as surrogate users. If cognitive models could interact with the same interfaces that users do, the models would be easier to develop and would be easier to apply as interface testers. This approach can be encapsulated as a cognitive model interface management system (CMIMS), which is analogous to and based on a u ...

Results 1 - 20 of 41 short listing

Prev Page 1 2 3 Next Page

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2003 ACM, Inc.

PORTAL THE ACM DIGITAL LIBRARY

> home > about > feedback > login
US Patent & Trademark Office

Try the *new* Portal design
Give us your opinion after using it.

Search Results

Search Results for: **[CAD and (file or object) and update and distributed and overwrite]**
Found **60** of **120,398 searched.**

Search within Results

GO > Advanced Search > Search Help/Tips

Sort by: Title Publication Publication Date Score Binder

Results 1 - 20 of 60 short listing

Prev Page 1 2 3 4 Next Page

1 Special issue on persistent object systems: Orthogonally persistent object systems 84%
Malcolm Atkinson , Ronald Morrison
The VLDB Journal — The International Journal on Very Large Data Bases July 1995
Volume 4 Issue 3
Persistent Application Systems (PASs) are of increasing social and economic importance. They have the potential to be long-lived, concurrently accessed, and consist of large bodies of data and programs. Typical examples of PASs are CAD/CAM systems, office automation, CASE tools, software engineering environments, and patient-care support systems in hospitals. Orthogonally persistent object systems are intended to provide improved support for the design, construction, maintenance, and operation o ...

2 PRIME—toward process-integrated modeling environments: 1 80%
Klaus Pohl , Klaus Weidenhaupt , Ralf Dömges , Peter Haumer , Matthias Jarke , Ralf Klamma
ACM Transactions on Software Engineering and Methodology (TOSEM) October 1999
Volume 8 Issue 4
Research in process-centered environments (PCEs) has focused on project management support and has neglected method guidance for the engineers performing the (software) engineering process. It has been dominated by the search for suitable process-modeling languages and enactment mechanisms. The consequences of process orientation on the computer-based engineering environments, i.e., the interactive tools used during process performance, have been studied much less. In this article, we prese ...

3 Cooperative transaction hierarchies: transaction support for design applications 80%
Marian H. Nodine , Stanley B. Zdonik
The VLDB Journal — The International Journal on Very Large Data Bases July 1992
Volume 1 Issue 1

Traditional atomic and nested transactions are not always well-suited to cooperative applications, such as design applications. Cooperative applications place requirements on the database that may conflict with the serializability requirement. They require transactions to be long, possibly nested, and able to interact with each other in a structured way. We define a transaction framework, called a *cooperative transaction hierarchy,* that allows us to relax the requirement for atomic, seria ...

4 Programming languages and object technologies: Concurrency control for distributed cooperative engineering applications 80%

João Coelho Garcia , Paulo Ferreira

Proceedings of the 2002 ACM symposium on Applied computing March 2002

Distributed cooperative engineering applications require consistent and long-term sharing of large volumes of data, which may cause conflicts due to concurrent read/write operations. Therefore designing concurrency control for underlying middleware systems is a difficult issue.Current transactional solutions, even if based on an optimistic approach, do not solve the problem because such applications access shared data for long periods of time performing a large number of read/write operations. T ...

5 Adaptive, fine-grained sharing in a client-server OODBMS: a callback-based approach 80%

Markos Zaharioudakis , Michael J. Carey , Michael J. Franklin

ACM Transactions on Database Systems (TODS) December 1997

Volume 22 Issue 4

For reasons of simplicity and communication efficiency, a number of existing object-oriented database management systems are based on page server architectures; data pages are their minimum unit of transfer and client caching. Despite their efficiency, page servers are often criticized as being too retrictive when it comes to concurrency, as existing systems use pages as the minimum locking unit as well. In this paper we show how to support object-level locking in a page-server context. Sev ...

6 A survey of current object-oriented databases 80%

Mansour Zand , Val Collins , Dale Caviness

ACM SIGMIS Database February 1995

Volume 26 Issue 1

Object-oriented concepts form a good basis for the data models required for next-generation database applications such as CAD/CAE/CASE/CAM systems, knowledge-based systems, multimedia, etc. Many object-oriented databases are available commercially or are being developed by industry or academic research facilities. This paper attempts to compare some of these products using fourteen criteria. The selected criteria are major factors required for the successful design of an object-oriented database ...

7 System facilities for CAD databases 77%

Charles M. Eastman

Proceedings of the seventeenth design automation conference on Design automation June 1980

In this paper, an attempt is made to lay out the special needs of design databases, as compared to the facilities provided in conventional database systems now commercially available. The paper starts from a point of commonality and focusses on the limitations and shortcomings commonly found in current database systems. It is impossible and unwise to make universal statements about DBMS capabilities. Instead, the goal is to identify those special features that, by their capability,

provide ...

8 A CODASYL CAD data base system 77%

Günther Zintl

Proceedings of the eighteenth design automation conference on Design automation June 1981

If in CAD systems a CODASYL data base system is used, one has to avoid some well known disadvantages. To achieve this, we defined for CAD needs in application programs a special high level data manipulation language (HLDML). With this concept the application programs get independent from the data base structure and they are able to work with a “many records at a time”- view. Other provisions reduced some overhead caused by the used data base system.

9 Supporting procedural constructs in existing SQL compilers 77%

Gene Fuh , Jyh-Herng Chow , Nelson Mattos , Brian Tran

Proceedings of the 1996 conference of the Centre for Advanced Studies on Collaborative research November 1996

The draft of the SQL/PSM standard denes a procedural extension to the existing SQL2 language. An essential part of this extension is the support of procedural constructs such as BEGIN/END blocks, local variables, assignment statements, conditional statements, and various forms of loops.Such an extension introduces new challenges to existing SQL compilers. Most SQL compilers exiting in the marketplace today were built based on the declarativeness of SQL. The question is how these procedural exten ...

10 Continuous program optimization: A case study 77%

Thomas Kistler , Michael Franz

ACM Transactions on Programming Languages and Systems (TOPLAS) July 2003 Volume 25 Issue 4

Much of the software in everyday operation is not making optimal use of the hardware on which it actually runs. Among the reasons for this discrepancy are hardware/software mismatches, modularization overheads introduced by software engineering considerations, and the inability of systems to adapt to users' behaviors.A solution to these problems is to delay code generation until load time. This is the earliest point at which a piece of software can be fine-tuned to the actual capabilities of the ...

11 PCLOS: a critical review 77%

A. Paepcke

ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages and applications September 1989 Volume 24 Issue 10

This paper uses the persistent object system PCLOS to survey some problems and benefits of object persistence. The system is analyzed along several relevant dimensions. PCLOS provides object persistence for an object-oriented language. The insights gained on desirable and detrimental components of the system are presented. The intent is to outline some of the expected and unexpected problems encountered in the construction of support for object persistence.

12 The design and implementation of K: a high-level knowledge-base programming language of OSAM*.KBMS 77%

Yuh-Ming Shyy , Javier Arroyo , Stanley Y.W. Su , Herman Lam
The VLDB Journal — The International Journal on Very Large Data Bases August 1996
Volume 5 Issue 3
The OSAM*.KBMS is a knowledge-base management system, or the so-called next-generation database management system, for non-traditional data/knowledge-intensive applications. In order to define, query, and manipulate a knowledge base, as well as to write codes to implement any application system, we have developed an object-oriented knowledge-base programming language called K to serve as the high-level interface of OSAM*.KBMS. This paper presents the design of K, its implementation, and its supp ...

13 Synchronization and recovery in a client-server storage system 77%
E. Panagos , A. Biliris
The VLDB Journal — The International Journal on Very Large Data Bases August 1997
Volume 6 Issue 3
Client-server object-oriented database management systems differ significantly from traditional centralized systems in terms of their architecture and the applications they target. In this paper, we present the client-server architecture of the EOS storage manager and we describe the concurrency control and recovery mechanisms it employs. EOS offers a semi-optimistic locking scheme based on the multi-granularity two-version two-phase locking protocol. Under this scheme, multiple concurrent reade ...

14 Loop optimization in register-transfer scheduling for DSP-systems 77%
G. Goossens , J. Vandewlle , H. De Man
Proceedings of the 1989 26th ACM/IEEE conference on Design automation conference June 1989
In this paper, we discuss a control-flow transformation called loop folding, during the scheduling of register-transfer code for DSP-systems. Loop folding is functionally equivalent to data-path pipelining. An iterative loop-folding procedure, implemented in the CATHEDRAL II compiler, is presented. This technique may significantly improve the utilization of parallel hardware, available in a data path.

15 State trees as structured finite state machines for user interfaces 77%
James Rumbaugh
Proceedings of the 1st annual ACM SIGGRAPH symposium on User Interface Software January 1988
State trees are a technique for specifying the control of an interactive system by organizing states into trees to define shared structure and behavior. The tree structure permits inheritance of state information, event traps, and entry and exit actions from states to substates, thereby sharing information and reducing the amount of code to be written. An interface can be restructured by moving entire subtrees as modules. State trees separate the recognition of commands from their implement ...

16 Chronological scheduling of transactions with temporal dependencies 77%
Dimitrios Georgakopoulos , Marek Rusinkiewicz , Witold Litwin
The VLDB Journal — The International Journal on Very Large Data Bases January 1994

Volume 3 Issue 1
Database applications often impose temporal dependencies between transactions that must be satisfied to preserve data consistency. The extant correctness criteria used to schedule the execution of concurrent transactions are either time independent or use strict, difficult to satisfy real-time constraints. On one end of the spectrum, serializability completely ignores time. On the other end, deadline scheduling approaches consider the outcome of each transaction execution correct only if the tra ...

17 Nested transactions in a combined IRS-DBMS architecture 77%
Hans-Joerg Schek
Proceedings of the 7th annual international ACM SIGIR conference on Research and development in information retrieval July 1984
The possibility to put an Information Retrieval System (IRS)on top of a Data Base Management System (DBMS) is investigated with respect to concurrency control and recovery. The simple mapping of one IRS transaction to one DBMS transaction is analyzed and found to be unsatisfactory. Therefore, the idea of generating a sequence of DBMS transactions for a single IRS transaction is discussed. This notion of "nested" transactions assumes standard concurrency control technology but is applied twice: i ...

18 Session 2: embedded system techniques (1): A near-optimal instruction scheduler for a tightly constrained, variable instruction set embedded processor 77%
Jack Liu , Fred Chow
Proceedings of the international conference on Compilers, architecture, and synthesis for embedded systems October 2002
This paper describes the implementation of an instruction scheduler that produces near-optimal results by efficiently enumerating all possible schedules. The scheduler is implemented in the context of an embedded processor that allows compile-time configuration of the instruction set to use for each program, where the need to adhere to the tight and irregular hardware constraints were the original motivation for using this approach. We present techniques that speed up the enumeration process by ...

19 Ray tracing vs. scan conversion: SaarCOR: a hardware architecture for ray tracing 77%
Jörg Schmittler , Ingo Wald , Philipp Slusallek
Proceedings of the conference on Graphics hardware 2002 September 2002
The ray tracing algorithmis well-known for its ability to generate high-quality images and its flexibility to support advanced rendering and lighting effects. Interactive ray tracing has been shown to work well on clusters of PCs and supercomputers but direct hardware support for ray tracing has been difficult to implement.In this paper, we present a new, scalable, modular, and highly efficient hardware architecture for real-time ray tracing. It achieves high performance with extremely low memor ...

20 The VMP multiprocessor: initial experience, refinements, and performance evaluation 77%
D. R. Cheriton , A. Gupta , P. D. Boyle , H. A. Goosen
ACM SIGARCH Computer Architecture News , Proceedings of the 15th Annual International Symposium on Computer architecture May 1988
Volume 16 Issue 2
VMP is an experimental multiprocessor being developed at Stanford University, suitable for high-performance workstations and server machines. Its primary novelty lies in the use of

software management of the per-processor caches and the design decisions in the cache and bus that make this approach feasible. The design and some uniprocessor trace-driven simulations indicating its performance have been reported previously. In this paper, we present our initial experience with the V ...

Results 1 - 20 of 60 short listing

Prev Page 1 2 3 4 Next Page

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2003 ACM, Inc.